

Το πρόβλημα της μέγιστης υποακολουθίας

Κώστας Άγγελος

2 Νοεμβρίου 2022

Περίληψη

Με τον όρο πολυπλοκότητα ορίζουμε την διαδικασία υπολογισμού ενός τύπου ο οποίος προσδιορίζει τον συνολικό χρόνο που απαιτείται για την εκτέλεση ενός συγκεκριμένου αλγορίθμου για είσοδο δεδομένων μεγέθους (n). Στην συγκεκριμένη εργασία αναλύονται τρεις αλγόριθμοι που λύνουν το πρόβλημα της μέγιστης υποακολουθίας. Για κάθε έναν ξεχωριστά υπολογίζουμε την πολυπλοκότητα, τονίζοντας έτσι την διαφορά σε ταχύτητα λόγω της επιλογής ενός ικανότερου αλγορίθμου.

1. Εισαγωγή

Το πρόβλημα ουσιαστικά είναι το εξής : Δεδομένου μια ακολουθία ακεραίων τιμών μας ζητείται να βρούμε την υποακολουθία με το μεγαλύτερο άθροισμα. Στην συγκεκριμένη αναφορά παρουσιάζονται τρεις αλγόριθμοι που λύνουν το πρόβλημα :

1) Brute Force Algorithm

Αποτελεί τον πιο απλό τρόπο υπολογισμού της υποακολουθίας γι' αυτό και έχει πολυπλοκότητα ίση με $O(n^3)$. Η ιδέα είναι ότι βρίσκουμε όλους τους πιθανούς sub arrays και εξάγουμε το μέγιστο άθροισμα.

2) Prefix Algorithm

Αποτελεί μια καλύτερη προσέγγιση με πολυπλοκότητα ίση με $O(n^2)$ καθώς υπολογίζει εξ αρχής τα αθροίσματα.

3) Kadane Algorithm

Αποτελεί τον πιο γρήγορο τρόπο επίλυσης του προβλήματος με πολυπλοκότητα ίση με $O(n)$.

2. Πειραματικά αποτελέσματα

Για την καταγραφή των πειραματικών αποτελεσμάτων χρησιμοποιήθηκε το εξής setup :

- IDE : PyCharm 2022.2.3
- Γλώσσα προγραμματισμού : Python 3.10.8
- Βιβλιοθήκη γραφικών : matplotlib

Χαρακτηριστικά υπολογιστή :

- CPU : AMD Ryzen 5 3400g
- RAM : 32 GB DDR4 (2933 MHz)

Παράμετροι πειραμάτων :

Για την μέτρηση του χρόνου εκτέλεσης των αλγορίθμων χρησιμοποιήθηκαν τυχαίοι αριθμοί με εύρος το $[-100,100]$. Για κάθε έναν αλγόριθμο χρησιμοποιήθηκε πλήθος από διάφορα N . Στην συγκεκριμένη αναφορά έχουμε: $N \in \{100,500,1.000,5.000\}$.

Παρακάτω παρουσιάζονται τα αποτελέσματα του χρόνου εκτέλεσης (σε Seconds) για διαφορετικό αριθμό από N .

Ένα sample της εξόδου του προγράμματος :

```
Brute Force Approach --> For N=100      Max sum is : 1254      Start Index : 2      End index : 75      time elapsed : 0.011622 seconds
Brute Force Approach --> For N=500      Max sum is : 1313      Start Index : 17     End index : 213     time elapsed : 1.191646 seconds
Brute Force Approach --> For N=1000     Max sum is : 918       Start Index : 906    End index : 965     time elapsed : 9.411697 seconds
Brute Force Approach --> For N=5000     Max sum is : 8825      Start Index : 171    End index : 2566    time elapsed : 1238.046363 seconds
Prefix Approach --> For N=100          Max sum is : 1254      Start Index : 2      End index : 75      time elapsed : 0.00049 seconds
Prefix Approach --> For N=500          Max sum is : 1313      Start Index : 17     End index : 213     time elapsed : 0.013653 seconds
Prefix Approach --> For N=1000         Max sum is : 918       Start Index : 906    End index : 965     time elapsed : 0.062066 seconds
Prefix Approach --> For N=5000         Max sum is : 8825      Start Index : 171    End index : 2566    time elapsed : 1.369096 seconds
Kadane Algorithm Approach --> For N=100  Max sum is : 1254      Start Index : 4      End index : 75      time elapsed : 5.7e-05 seconds
Kadane Algorithm Approach --> For N=500  Max sum is : 1313      Start Index : 17     End index : 213     time elapsed : 0.000181 seconds
Kadane Algorithm Approach --> For N=1000  Max sum is : 918       Start Index : 906    End index : 965     time elapsed : 0.000331 seconds
Kadane Algorithm Approach --> For N=5000  Max sum is : 8825      Start Index : 171    End index : 2566    time elapsed : 0.001749 seconds
```

Εικόνα 1. Output κυρίως προγράμματος

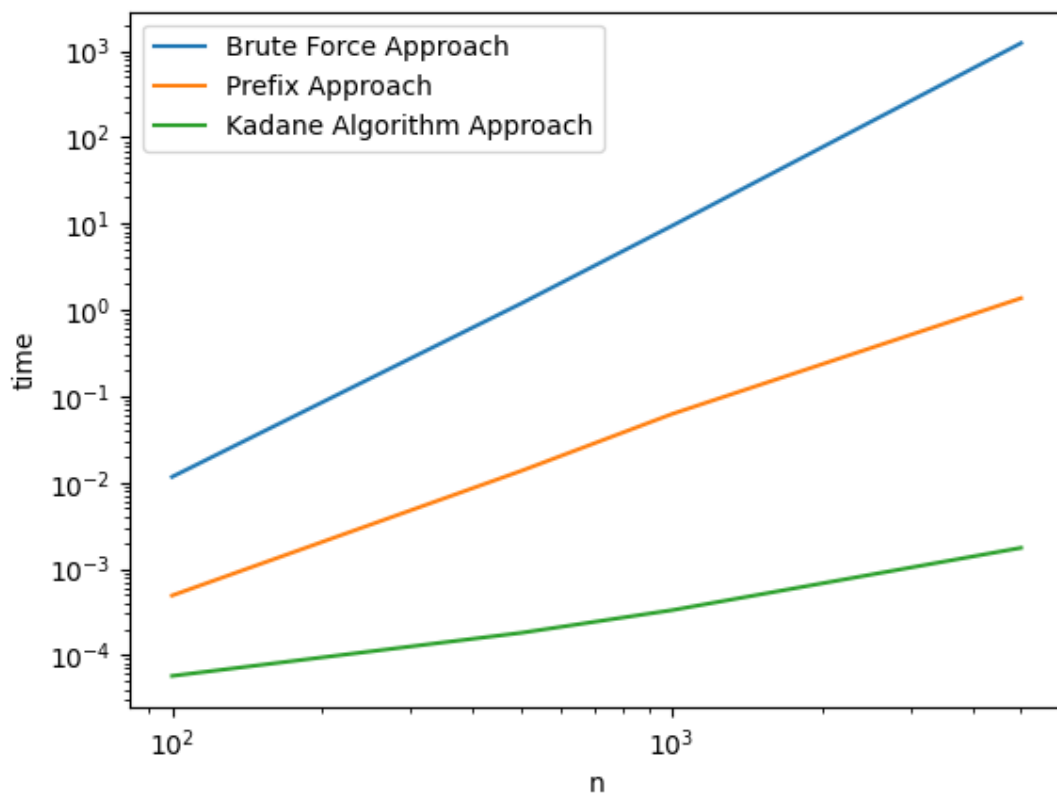
```
PS C:\Users\Leader\Desktop\alco_assignment_1> py tests.py
...
-----
Ran 3 tests in 0.000s

OK
```

Εικόνα 2. Output των tests

	N=100	N=500	N=1.000	N=5.000
Brute Force Algorithm	0.011622	1.191646	9.411697	1238.046363
Prefix Algorithm	49e-4	0.013653	0.062066	1.369096
Kadane Algorithm	5.7e-05	181e-4	331e-4	0.001749

Πίνακας 1. Πίνακας αποτελεσμάτων



Εικόνα 3. Οπτικοποίηση αποτελεσμάτων (matplotlib)

2. Συμπεράσματα

Από το συγκεκριμένο πρόβλημα αλλά και τις μετρήσεις που εκτελέσαμε φαίνεται η χρησιμότητα σε χρόνο και χώρο από την επιλογή ενός ικανού και γρηγορότερου αλγόριθμου σε σχέση με απλούστερες υλοποιήσεις.

Συγκεκριμένα υπάρχει τεράστια διαφορά στο performance ιδίως εάν χρησιμοποιήσουμε μεγάλο αριθμό σε N . Για παράδειγμα για $N=10.000$ ο brute force μπορεί να διαρκέσει έως και 3.5 ώρες μέχρι να βρει λύση για το πρόβλημα (ανάλογα πάντα και με τα χαρακτηριστικά στα οποία διεξάγεται το πείραμα). Για την συγκεκριμένη αναφορά το εύρος του N επιλέχθηκε με γνώμονα να υπάρχει ισορροπία μεταξύ ταχύτητας και efficiency.